

PRZEMYSŁAW ŻARNECKI

---

# PYTHON 02

# MAŁE PRZYPOMNIENIE

# PYCHARM – SPOSÓB NA ZDOBYCIE LICENCJI:

JET  
BRAINS

Developer ToolsTeam ToolsEducationSolutionsSupportStore

Coming in 2023.3New UIWhat's NewFeaturesLearnPricingDownload

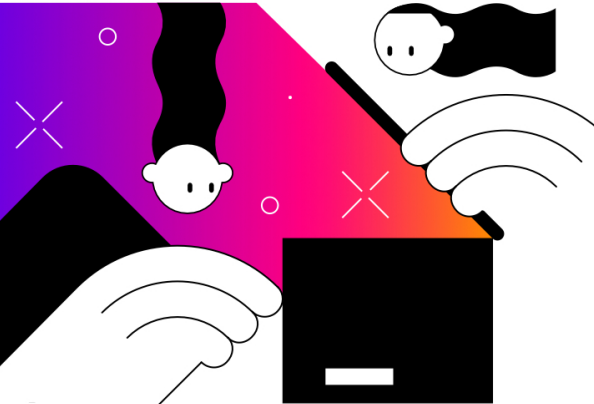
PyCharm

Free License Programs

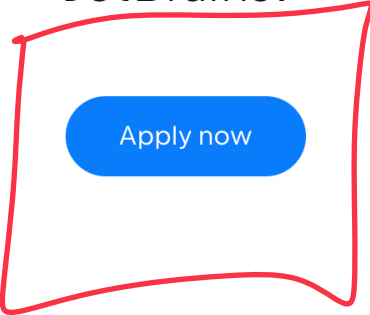
Academic LicensingOpen SourceUser GroupsEvents PartnershipDeveloper Recognition

✔ May be renewed free of charge as long as you are a student or a teacher.

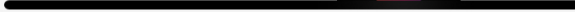
✘ May not be shared with any third parties.



Get free access to all developer tools from JetBrains!



3



JET  
BRAINS

Developer ToolsTeam ToolsEducationSolutionsSupportStore

JetBrains Academy

Find your way in learning or teaching computer science

FOR LEARNERS

Programming languages  
Select a language and try different approaches to learning it

Career fields  
Explore careers and see where programming could take you

University relations  
Study offline with academic programs

Internships  
Apply for internships and flexible jobs for students

FOR EDUCATORS

Teaching with JetBrains IDEs  
Create courses and share your knowledge

Kotlin for education  
Teach a wide range of Kotlin courses

FOR TEAMS

Professional development  
Ensure your team has up-to-date technical skills

FREE LICENSES

For students and teachers  
JetBrains IDEs for individual academic use

For educational institutions  
JetBrains IDEs and team tools for classroom use

For bootcamps and courses  
JetBrains IDEs for your students

Full-fledged Professional or Free Community

JetBrains tools are FREE for education!  
Enjoy a professional developer experience using industry-leading tools to learn, teach, and collaborate.

Get started now

2

### W SUMIE NIECO PRZYPOMNIENIA

Nzeczy  
równie  
sobie  
w jednej  
linii

punkty = 60



if punkty >= 65:

print("Ocena pozytywna")



Jak coś  
podręczne to  
wzięcie

### W SUMIE NIECO PRZYPOMNIENIA

`punkty = 60` *ZMIENNA*

if punkty >= 65:

print("Ocena pozytywna")


*!* `punkty = int(input("Podaj ilość punktów"))`

if punkty >= 65:

print("Ocena pozytywna")

## W SUMIE NIECO PRZYPOMNIENIA

```
punkty = 60  
  
if punkty >= 65:  
    print("Ocena pozytywna")  
else:  
    print("Ocena negatywna")
```



If; else są  
sobie równe.  
więc tu nie ma  
wzłąć

# W SUMIE NIECO PRZYPOMNIENIA

```
if punkty >= 90:
```

```
    print("ocena 5")
```

```
elif punkty >= 80:
```

```
    print("ocena 4")
```

```
elif punkty >= 70:
```

```
    print("ocena 3")
```

```
elif punkty >= 65:
```

```
    print("ocena 2")
```

```
else:
```

```
    print("ocena 1")
```

## FUNKCJE ZAGNIEŻDŻONE – WARTO STOSOWAĆ. UWAGA NA WCIECIA

if warunek1:                    #"zewnętrzna" instrukcja if

    print("true")

jeśli będzie True → wykona "2" if

    if zagniezdzony\_warunek:   #wewnętrzna instrukcja if

        print("yes")

    else:                       #wewnętrzna instrukcja else

        print("no")

else:                       #wewnętrzna instrukcja else

    print("false")

wykona jeżeli /  
będzie false

# FOR VS WHILE

	Pętla `for`	Pętla `while`
Wykorzystanie	Do iteracji po sekwencji	Do iteracji z warunkiem
Warunek stopu	Z góry określony (ilość iteracji)	Zależy od spełnienia warunku
Przykład	<code>`for i in range(5):`</code>	<code>`while x &lt; 10:`</code>
Kontrola	Bardziej kontrolowana przez język	Może wymagać bardziej ręcznej kontroli
Czytelność	Zazwyczaj bardziej czytelna, gdy iterujemy po kolekcji	Czasami bardziej czytelna, gdy warunek jest złożony
Potencjalne nieskończone wykonanie	Bardziej bezpieczna, trudniej o nieskończoną pętlę	Potencjalnie bardziej ryzykowna, łatwiej o nieskończoną pętlę

## FOR – DEFINICJE

for element in sekwencja:

# Ciało pętli

# Wykonuje się dla każdego elementu w sekwencji

- element to zmienna tymczasowa, która przyjmuje wartość każdego elementu w sekwencji podczas iteracji.
- sekwencja to dowolny obiekt iterowalny, taki jak lista, krotka, napis, itp.:

```
lista = [1, 2, 3, 4, 5]
```

```
for element in lista:
```

```
    print(element)
```

Ten przykład wydrukuje liczby od 1 do 5, ponieważ pętla for iteruje przez elementy listy lista i wyświetla je na ekranie

Pętla for to jedno z podstawowych narzędzi do powtarzania operacji w python.

## FOR PRZYKŁADY

---

```
for i in range(1,11):  
    print(i)
```

1

2

Po instrukcji for, Python wie, że ma wykonać następujący po niej kod, wielokrotnie.

3

4

1. Za każdym powtórzeniem, zmienna 'i', będzie zawierać inną liczbę. Zmienna 'i', oczywiście może się nazywać inaczej.

5

6

7

2. Wartości jakie będzie przyjmować zmienna 'i', będą w zakresie określonym przez funkcję range. Od 1 do 11.

8

9

10

3. Funkcja 'print', odwołuje się do zmiennej i, po czym wyświetla jej zawartość

\

|

## FOR PRZYKŁADY

---

```
lista = [1,2,'abc',3,4]
```

```
for i in lista:  
    print (i)
```

1

2

abc

3

4

---

```
tekst = "Analityk"  
  
for ttt in tekst:  
    print (ttt)
```

A  
n  
a  
l  
i  
t  
y  
k

# WHILE – DEFINICJA

---

Pętla `while` w języku Python to struktura kontrolująca powtarzanie określonego bloku kodu tak długo, jak dany warunek jest spełniony. Jest to narzędzie programistyczne, które umożliwia iteracyjne wykonywanie operacji, z uwzględnieniem warunku zakończenia, który decyduje, kiedy pętla przestaje działać."

```
licznik = 0
```

```
while licznik < 5:
```

```
    print("Aktualna wartość licznika:", licznik)
```

```
    licznik += 1
```

– Zwiększenie zmiennej o 1 w każdym obiegu pętli

# WHILE

```
krok = 0
```

```
while krok < 10:  
    print (krok)
```

```
    krok = krok + 1
```

inny sposób zwiększania

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

# WHILE

---

```
while True:  
    print("hi")
```

hi  
hi  
hi  
hi  
hi  
hi  
hi  
hi  
hi

Co można?

- czekać na zdarzenia użytkownika
- sprawdzać czy pojawił się jakiś plik
- sprawdzać czy pojawiły się jakieś dane
- i wiele innych

# WHILE

---

```
lista = ['A', 1, 2, 'abc', 'x']  
  
for element in lista:  
    if element == 2:  
        break  
    else:  
        print(element)  
  
print ("Pętla została zakończona")
```

A

1

Pętla została zakończona

---

## WHILE – BREAK MOŻNA STOSOWAĆ TEŻ W FOR

---

```
lista = ['A', 1, 2, 'abc', 'x']  
  
for element in lista:  
    if element == 2:  
        break  
    else:  
        print(element)  
  
print ("Pętla została zakończona")
```

A

1

Pętla została zakończona

---

# DEFINIOWANIE FUNKCJI W PYTHON

---

```
def nazwa_funkcji(argumenty):  
  
    # Ciało funkcji  
  
    # Możesz wykonywać operacje tutaj  
  
    return wynik # Opcjonalnie zwracasz wynik
```

Gdzie:

- nazwa\_funkcji to nazwa twojej funkcji.
- argumenty to lista argumentów, które funkcja może przyjmować. Mogą być one opcjonalne.
- return to opcjonalny operator, który pozwala na zwrócenie wartości z funkcji.

# DEFINIOWANIE FUNKCJI W PYTHON

---

Przykład definicji i użycia prostej funkcji w Pythonie:

```
def witaj(imie):
```

```
    powitanie = "Witaj, " + imie + "!"
```

```
    return powitanie
```

```
# Wywołanie funkcji i wydrukowanie wyniku
```

```
print(witaj("Jan")) # Wynik: "Witaj, Jan!"
```

# DEFINIOWANIE FUNKCJI W PYTHON

---

```
def say_hello():
```

```
    # Blok należący do funkcji.
```

```
    print('hello world')
```

```
# Koniec funkcji.
```

```
say_hello() # Wywołanie funkcji.
```

```
say_hello() # Ponowne wywołanie funkcji.
```